17th Australasian Fluid Mechanics Conference
Auckland, New Zealand
5-9 December 2010

Meshing challenges in complicated FSI flows: Aortic dissection and coiled aneurysms

K R Moyle[1,2] and Y Ventikos[2]

[1]Department of Engineering Science
University of Auckland, Auckland, New Zealand

[2]Faculty of Engineering Science
University of Oxford, Oxford, England

Abstract

Bioengineering and the development of endovascular techniques such as stenting and coiling present challenges to computational modelling, mostly in the area of meshing. We have developed a suite of software tools able to create high-quality meshes around complicated geometries, as well as locally remeshing and repairing elements undergoing large deformations. The theory and possible applications are described below.

Introduction

As the capabilities and speed of numerical simulations of fluid flow continue to increase, the bottleneck in the creation of high-quality solutions has been the meshing technologies available. Whether this be due to the wide variation in length scales required to mesh a stented aneurysm, or the need for highly deformable meshes to model the motion of an aortic dissection membrane or heart valve, the software suite can help in the creation and maintenance of unstructured high-quality meshes.

Methodology

The complicated and arbitrary geometries encountered in bioengineering problems have meant that we concentrate predominantly on unstructured tetrahedral meshes, though some of the methodologies developed (such as the fast mesh motion) can be applied to any element topology.

Fast mesh motion

The ability of a computational mesh to model a fluid domain which has moving boundaries requires the internal fluid mesh to move also. Traditionally, the mesh itself has been used to transport information on the boundary position into the fluid domain, but this is an unnecessarily expensive process. Instead, we use a linear interpolation between boundary nodes into regions of the mesh that are local to those nodes. Locality is defined using a Delaunay tetrahedralisation between the boundary nodes as shown in figure 1, and detailed further in [1,2].

Dynamic remeshing

In situations involving extensive domain deformation, especially shear deformation, local element quality may drop to levels unsuitable for simulation. Remeshing the entire domain is an expensive option, and can introduce interpolation errors between the old and new meshes. We have developed a technique that classifies low-quality elements into types according to the manner best suited to their repair and removal, as shown in figure 2.

The type of each element and heuristic thresholds for quality then determine the new connectivity for the nodes and surrounding elements. The use of edge-splitting and edge-collapse to midpoints helps to reduce interpolation errors during the procedure.

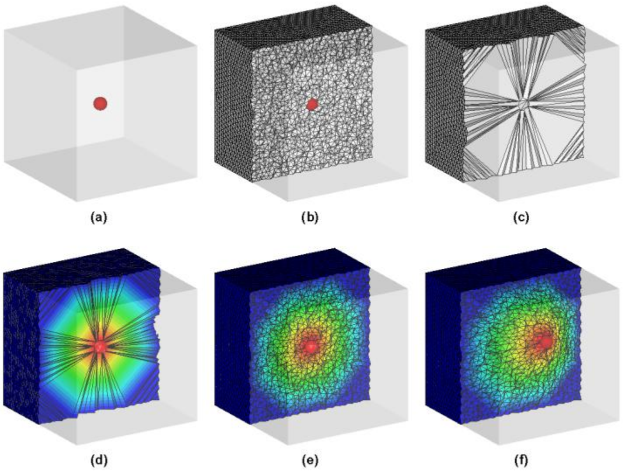

Figure 1: (a) A sphere in box, where the motion of the sphere is prescribed and the box filled (b) with tetrahedra. A Delaunay tetrahedralisation (c) is created between boundary nodes on the two surfaces, and values of known displacement at these boundary nodes (d) interpolated (e) onto the internal mesh nodes. The internal nodes are then moved (f) according to these interpolated values.

Type I elements contain no short edges, but two crossed diagonal edges over a quadrilateral footprint. These are removed by splitting both crossed diagonals, and collapsing the short edge formed from their midpoints, as shown in figure 3.

Type II elements are similar to type I in having crossed diagonals, but have one corner within the short-edge tolerance of the opposite edge’s midpoint. They are removed by splitting only the opposite edge, and collapsing the short edge formed between the near corner and the new midpoint, as shown in figure 4.

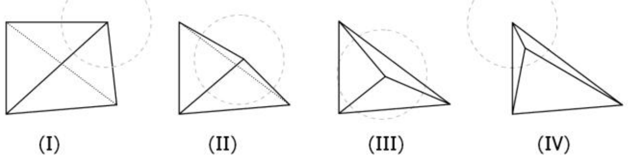

Figure 2: Type classification of low-quality tetrahedra. The dashed circle shows the limits of the short edge criterion used by Anderson et al [3] for edge collapse; it can be seen that only type IV elements would be removed using such a threshold.

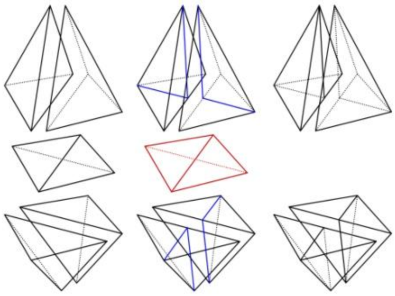

Figure 3: Repair of type I elements.

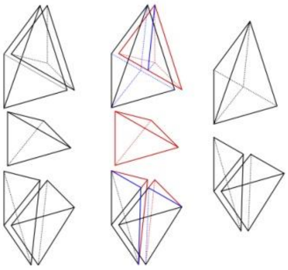

Figure 4: Repair of type II elements.

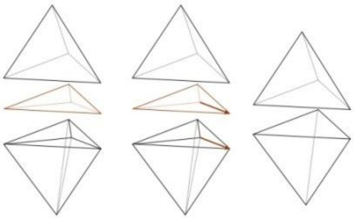

Figure 5: Repair of type III elements.

Type III elements have a triangular footprint, and are removed by collapsing the shortest edge between the central vertex and any of the corners, as shown in figure 5. Type IV elements are removed implicitly using standard thresholds of local edge length proposed in [3], where edges less than half the average length are collapsed, and more than 1.5 times as long are split.

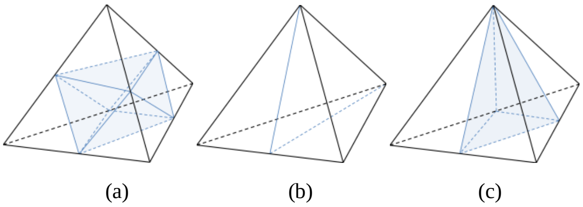

Figure 6: Effect of adaptive refinement subdivision strategy on tetrahedra (a) internal to the parent element, or external with (b) edge or (c) face split. Internal elements have a quality similar to their parent element, external elements generally have a lower quality owing to the combination of split and un-split edges.

Adaptive refinement

Adaptive refinement routines are built from concepts similar to the edge splitting described in the previous section, but are centred instead around whole element division, as shown in figure 6. Care is taken in the routines to record the division history, especially at the interface between divided and undivided elements to minimise the negative effect on element quality of the division process. This means that the two external elements show in figure 6 would have their long edges (those unconnected to the divided element) split before the short edges (those created by splitting the parent element) are split. Thus, the overall mesh quality after generations of adaptive refinement will be as close as possible to that of the original parent mesh.

Elements or regions of the mesh to refine can be selected based on solution variables, such as gradients or vorticity, or on proximity to a region of interest (ROI), such as a stent or coil.

Mesh intersection and implantation

Intersecting two meshed geometries requires three steps: (1) the location of nodes of the inner mesh within the outer, (2) the identification of nodes and elements of the outer mesh within the inner, (3) the removal of these outer mesh elements, and (4) the creation of new surfaces in the outer mesh, and ensuring their conformity to the surface of the inner mesh. This process is illustrated in figure 7.

Contact warning

Contact models are used in FSI in situations involving wall collisions, sliding and cleaving motions. Such models usually entail complicated and computationally expensive routines, which may only be usefully employed when the surfaces of interest are actually in contact with one another. The contact warning tool is simply a measure of the shortest distances between two surfaces within the model, achieved by combining a Delaunay mesh between the model surfaces and a directionally-aware algorithm for distance measurement. This measurement is illustrated in figures 8-10.

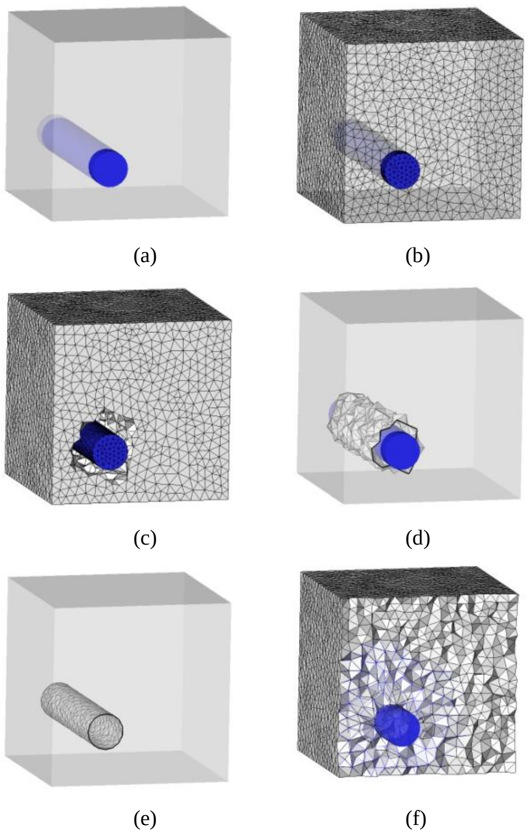

Figure 7: Steps of mesh intersection and implantation (a,b) original geometries are have independent meshes (c) the elements of the outer mesh that are contained within the inner mesh are removed (d) a surface is created and then (e) moved to conform to the surface of the inner object, in such a way as (f) the global mesh quality is not compromised.

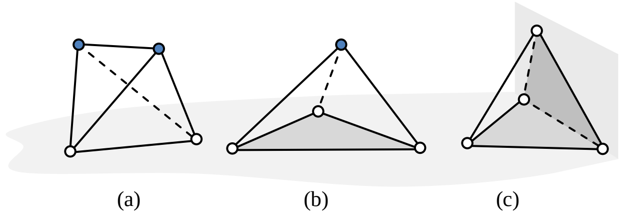

Figure 8: Classification of contact tetrahedra as (a) two-two, (b) three-one and (c) four according to the number of nodes on each separate contact surface.

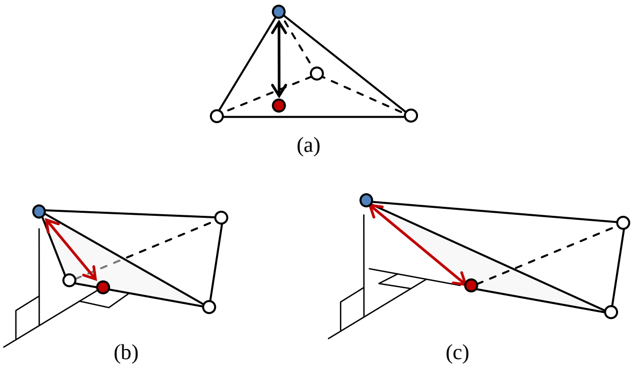

Figure 9: Possible arrangements of distance calculation based on the three-one tetrahedra. (a) The element is acute, and the nearest distance is normal to the three-node face, (b) the element is obtuse but the face is acute, and the nearest distance is normal to the base edge, and (c) the element and the face are acute, and the shortest distance is that between the nearest two nodes.

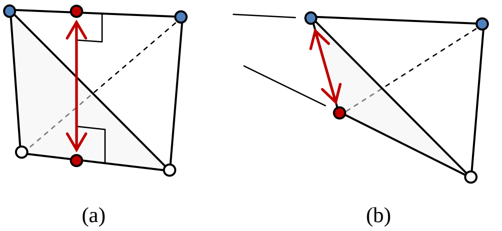

Figure 10: Possible arrangements of the two-two tetrahedron. In cases where the nearest distance between the skew line segments (the edges lying on each surface) occurs at a point on the line segments (a), then the skew distance is used. If not (b) then the shortest edge spanning the surfaces is used instead.

Testing and results

Each of the modules described has been subject to conceptual and, where appropriate, speed testing.

Fast mesh motion

Speed tests concerning the fast mesh motion method were performed, and the results compared to the standard pseudo-elasticity method used in many commercial solvers. Four different levels of mesh density were tested, each with and without a rigid layer around the boundaries. The results of these tests are shown in table 1, where the advantage of the fast Delaunay method is clearly seen in the three orders of magnitude speed increase gained over the traditional pseudo-elasticity method.

Dynamic remeshing

The local dynamic remeshing tools were tested and optimised on several geometries, including a box and membrane, a ball in box, and distorting cubes. Details of these tests can be found in [1]. Optimised parameters for edge reconnection, collapse and splitting as well as parameters for the method of removal for degenerate elements according to their classification (figure 2) are shown in table 2. Figure 11 shows the results from a translating sphere in a box, using the local remeshing algorithm.

model	Ad	A	Bd	B	Cd	C	Dd	D
volume nodes	10203		32102		72655		157887	
volume elements	52774		175181		403054		896185	
boundary nodes	2700		5704		10854		16947	
boundary elements	5392		11400		21700		33886	
Delaunay nodes	5400	2700	11408	5704	21708	10854	33894	16947
Delaunay elements	25273	7811	52450	16497	99141	31549	154615	49462
total Delaunay setup time (s)	2.21	1.20	6.14	3.47	14.0	8.47	28.9	19.2
computational timestep: elastic (s)	4.421		15.16		35.15		78.69	
computational timestep: Delaunay (s)	1.25e-3	2.22e-3	6.66e-3	5.65e-3	11.6e-3	11.5e-3	37.7e-3	30.8e-3
speed increase factor	3537	1989	2273	2682	3013	3046	2082	2549

Table 1: Results of speed comparisons between the fast Delaunay-based method and a commercial pseudo-elasticity solver (CFD-ACE+, ESI Group, Paris, France).

Measure	Limit	Classification
Edge-to-edge distance	< 0.18	Type I
Node-to-edge distance	< 0.21	Type II
Equi-angle skew	< 0.2, real node-to-face	Type III
Shortest edge fraction	< 0.2	Type IV

Table 2: Optimised limits for the removal method for each type of degenerate tetrahedra. Reproduced from [1].

Adaptive refinement and mesh intersection

The combination of adaptive mesh refinement routines and the mesh intersection methodology was used to create a conformal mesh around the coil wire inserted into an aneurysm. The overall geometry and selected cross-sections of the resulting mesh are shown in figure 12.

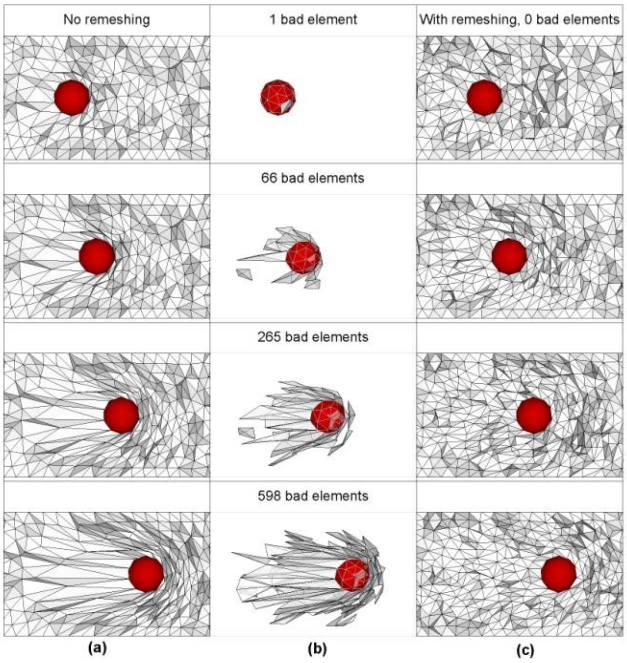

Figure 11: Effect of local remeshing on the number of “bad” elements (EAS < 0.2) in the mesh of a sphere moving along a channel. (a,b) The pseudo-elastic deformation leaves 1, 66, 265 and 598 poorly shaped elements as the sphere is moved to the right, (c) shows the mesh with all low quality elements removed using the local remeshing routines.

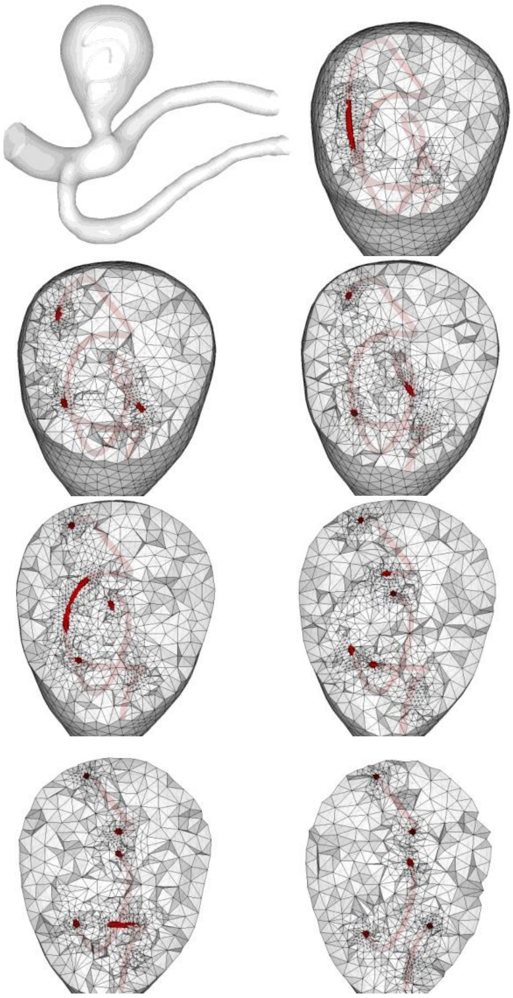

Figure 12: The intersection and adaptive refinement of an aneurysm mesh around a coil wire inserted into it.

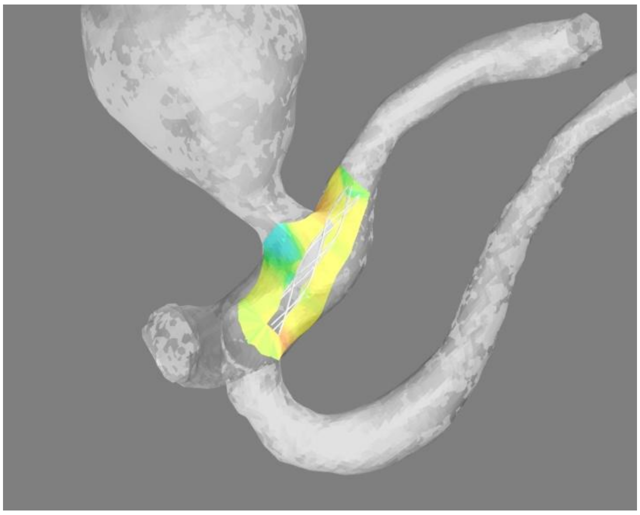

Figure 13: The nearest surface distances between an implanted but unexpanded stent and an aneurysm geometry. The stent would be expanded using a balloon, and the interaction with the wall modelled when near enough to activate the contact model.

Contact warning

Capabilities of the contact warning method are shown in figure 13, where the shortest distances between an unexpanded stent and the walls of an aneurismal vessel are shown. The efficacy of using a contact model warning system is highly dependent on the problem to be modelled and the speed of the contact model processor. For complicated and expensive contact models, the use of this system could be expected to improve the running time of the total program; for situations in which contact is achieved throughout the simulation, there is no advantage.

Implementation and availability

The routines and modules described here have been developed externally to any specific commercial solver or mesh generation tool, and are designed to be easy to implement using the user-specified subroutine options available in many commercial packages. Researchers fortunate enough to have source code access to solvers are also able to include these modules in their compilations. We must emphasise that this code references other open-source software [Tetgen, Han Si, http://tetgen.berlios.de] and may only be used under the conditions listed by these contributing sources. More information and public downloads will be made available at the conference.

Conclusions

We have described new software developed specifically for complicated meshing and remeshing applications, including fast mesh motion, dynamic remeshing, mesh intersection and contact warning. We will make this software publicly available at the conference.

References

[1] Moyle, K. R., & Ventikos, Y., Local remeshing for large amplitude grid deformation, *J. Comp. Phy* 227, 2008, 2781-2793.

[2] Liu, X., Qin, N., & Xia, H., Fast dynamic grid deformation based on Delaunay graph mapping, *J. Comp. Phys* 211, 2006, 405-423

[3] Anderson, A., Zheng, X., & Cristini, V., Adaptice unstructured volume remeshing – I: The method, *J. Comp. Phys.* **208**, 2005, 616–625

[4] Moyle, K.R., Ford, M.D., & Steinman, D.A., Meshing techniques for aneurysm interventional modelling, World Congress of Biomechanics, Munich, July 2006